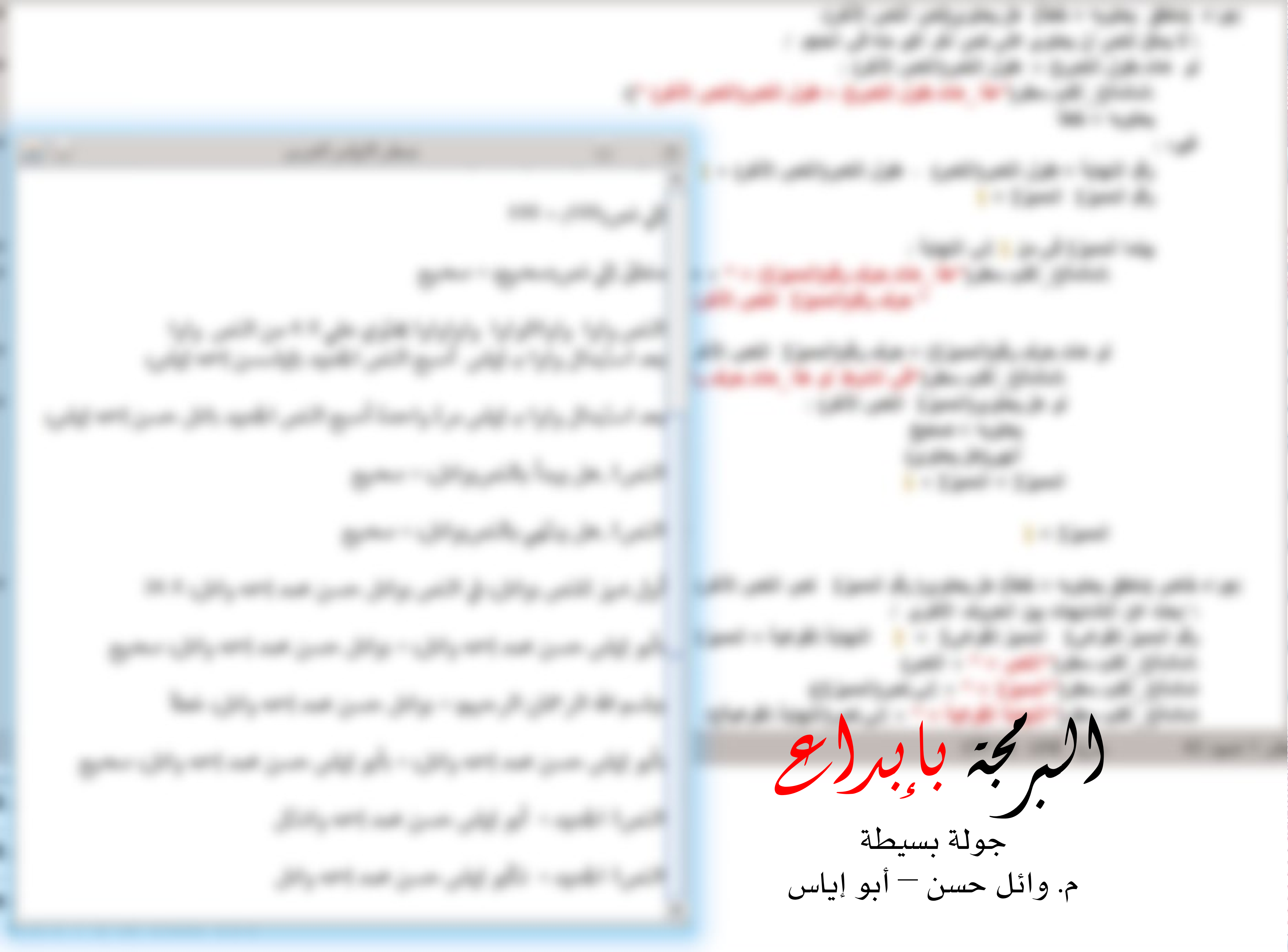
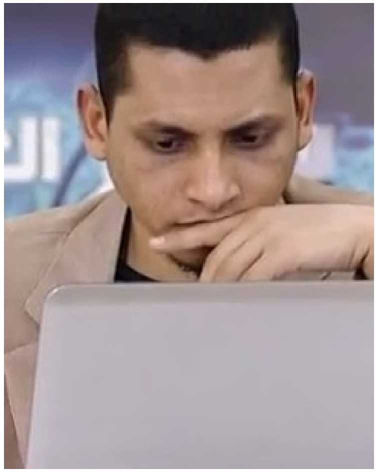


البرمجة بإبداع

جولة بسيطة

م. وائل حسن – أبو إياس



من أنا ؟

- الاسم: وائل حسن محمدعلي عبد المنعم.
- الكنية: أبو إياس.
- الميلاد: 22 أكتوبر 1987 م.
- خريج 2010 من كلية الهندسة و التكنولوجيا بأسوان (قسم كهرباء - شعبة حاسبات و نظم).
- التخصص الحالي: مهندس برمجيات في مجال تصميم و بناء لغات البرمجة. (بناء لغات البرمجة يتضمن تصميم و بناء المترجمات compilers و المفسرات interpreters و غيرهما من الأدوات الأخرى).
- مدونتي الشخصية:

  afkar-abo-eyas.blogspot.com
- مدونتي العلمية:

  abo-eyas.blogspot.com
- البريد الإلكتروني: wael_hasan_87@yahoo.com

النقاط:



- نبذة عامة عن مشروع "البرمجة بإبداع"،
- نبذة عن تصميم لغة البرمجة العربية "إبداع"،
- المرحلة الحالية من المشروع،
- ماذا سيحدث في المراحل القادمة ؟،
- كيفية المساعدة لإنجاح المشروع.

أهداف المشروع:

• تأليف و ترجمة المُؤلّفات التي تخص العلوم البرمجية الأساسية إلى اللغة العربية:

العلوم المتعلقة بتصميم لغات البرمجة و إنتاج المُترجِمات compilers و المُفسِّرات interpreters و المُنقِّحات debuggers و كافة الأدوات البرمجية الخاصة بها، وصولاً إلى إنتاج بيئات البرمجة المتكاملة IDE، بل و وصولاً إلى ترجمة كل ما يتعلق بتصميم و إنتاج نظم التشغيل المختلفة.

- موسوعة الألسن:

ضم تلخيصاتٍ للعديد من لغات البرمجة التي تعلمتُها في كتابٍ واحدٍ، بحيث يغني حامله عن عشرات الكتب الأخري.

http://alson-enc.blogspot.com/

- إنتاج مُنتَجاتٍ عربيةٍ تخص الأدوات الرئيسة فى المجال البرمجى، بدءاً من لغات البرمجة العربية بمختلف أنواعها و انتهاءً بنظم التشغيل العربية بمختلف أنواعها.

- لغة البرمجة العربية الإحترافية: إبداع

http://ebda3lang.blogspot.com/

الخطة الزمنية للمشروع:

أطوار المشروع:

سيكون للمشروع طَوْران من أطوار الحياة، الطَوْر الأول هو: طَوْر العمل الفردي، و الطَوْر الثاني هو: طَوْر العمل المُؤسِّسي الجماعي،

- الطَوْر الفردي

المرحلة الأولي من حياة المشروع، و فيها ستكون مُنتجاته مُغلقة المصدر و سيكون العمل علي تطويرها فردياً تماماً؛ حيث سأكونُ المُبرمِجَ الوحيد فيه، و الوحيدَ الذي يَتعامل مع التساؤلات و/أو الإقتراحات و/أو الدعايا و/أو النقاشات المُختلفة لمن يهتمون بالمشروع (سلباً أو إيجاباً).

تتمة الخطة الزمنية للمشروع:

- الطَوْر المُؤسِّسي:

و فيه سيتم تحويل المشروع من عملٍ فرديٍ إلي مُؤسِّسةٍ لها كيانٌ خاص، و سيتم قبول مُشارَكات من يَرغب في تطوير الاكواد المصدرية لمُنتجات المؤسسة، مع الأخذ في الاعتبار أن أسماء المُنتَجات و شعاراتها سيتم اعتبارها علاماتٍ تجارية (تماماً كما يحدث في حالة نواة اللينُوكس linux kernel).

أما التمويل فسيَكون مُعتمِداً علي الطرق التالية:

- التبرعات
- شراء المنشورات الخاصة بالمشروع، مثل كتب تعليم البرمجة بلغة إبداع، و كتب تعلم التعامل مع المُفسِّر و المُترجِم و بيئة البرمجة المتكاملة و غيرهن من المُنتَجات الأخري.
- بيع خدمات التنزيل installing و الصيانة maintaining.
- بيع خدمات التدريب للأفراد، و للمُؤسِّسات.

تتمة الخطة الزمنية للمشروع:

موعد الإنتقال من الطَوْر الفردي إلي الطَوْر المُؤسِّسي:

هذا الأمر يعتمد بشكلٍ كاملٍ علي ما يتوفر من تمويل،

كيفية أخذ القرار في المُؤسِّسة فيما بعد:

أخذ القرارات في كل مشروعٍ من المشاريع المُختلِفة سيكون بمنهج التشاور الواضح بين أفراد مُجتمَع المُطوِّرين له، مثلما يَحدث في مُجتمَع نواة اللينوكس (بل ربما نعتمد علي ذات الآليات التي يعتمدون عليها في عملهم).

هل ستقتصر المُؤسِّسة علي البرمجيات فقط في المستقبل ؟

هناك نيةٌ صادقةٌ لَدَيَّ لتحويل المؤسسة إلي ما يُشبِه شركة apple في عالم الحوسبة، بحيث تكون قادرةً علي إنتاج أنظمةٍ مُتكامِلةٍ بكل ما تشمله من برمجياتٍ و عَتادٍ صلبٍ hardware تعمل عليه تلك البرمجيات.

مبادئ تصميم إبداع:

1 - استخدام الكلمات و الرموز المألوفة، و في وظائف تشبه وظيفتها في الحياة العادية:

+ - × ÷ ^ : ، { } ( )

2- تحويل النص البرمجى إلى ما يقارب الوصف العادى، حتى أن أى شخص غير متخصص يمكنه فهم الكثير بمجرد قراءة البرنامج المكتوب بإبداع. و لكن مع المحافظة على البعد عن الحشو.

3- مزج التعبيرات و المكونات البرمجية المتقاربة فى مكونٍ أو تعبيرٍ واحد أشمل و أعم؛ لتكون كذلك سهلة التعلم و الإستيعاب. و لتفادى البلبلة و الخلط الناجمين عن تشابه المكونات.

صنف = Struct + class + Interface

إجراء = Delegate + procedure + function

الجدول = array + hashtable

4- الإستغناء عن أى مكونٍ أو تعبيرٍ مربك بطبعه، مثل المؤشرات pointers.

```
void CvtImg2Dta(IplImage *img, double *slice)
{
    for( int y=0; y< img->height; y++ )// taking care of rows indexing
    {
        for( int x=0; x< img->width; x++ )// taking care of columns indexing
        {
            slice[img->nChannels*x+y*img->width*img->nChannels] =
                    (double)img->imageData[img->nChannels*x+y*img->widthStep]/255.0;
            if(img->nChannels==3)
            {
                slice[(3*x)+(3*y*img->width) +1] =
                    (double)img->imageData[(3*x)+(y*img->widthStep) +1]/255.0;
                slice[(3*x)+(3*y*img->width) +2] =
                    (double)img->imageData[(3*x)+(y*img->widthStep) +2]/255.0;
            }
        }
    }
}
```

5- جمع الأفضل من كل لغة برمجية:

- كثير من شكل الـpython
- بعض مواصفات الأصناف في الـjava
- فكرة التفويض من الـC#
- الـhashtable من الـpython

6- مساعدة المُبرمِج على إنتاج كود آمن عن طريق:

- معالجة الإستثناءات exception handling.
- تعبيرات و مكونات قوية تساعده على التغلب على الأجزاء الصعبة من مهمة التكويد و توفر عليه العناء. مثل مبدأ التوارث inheritance الذي يعطى المبرمج القدرة على الإستفادة من جهد الآخرين و إنتاجهم الذى تم التأكد من أمنه.

هيكل برنامج إبداع:

يتكون كل برنامجٍ مكتوبٍ بإبداع من:

1 - ملفٍ رئيس:

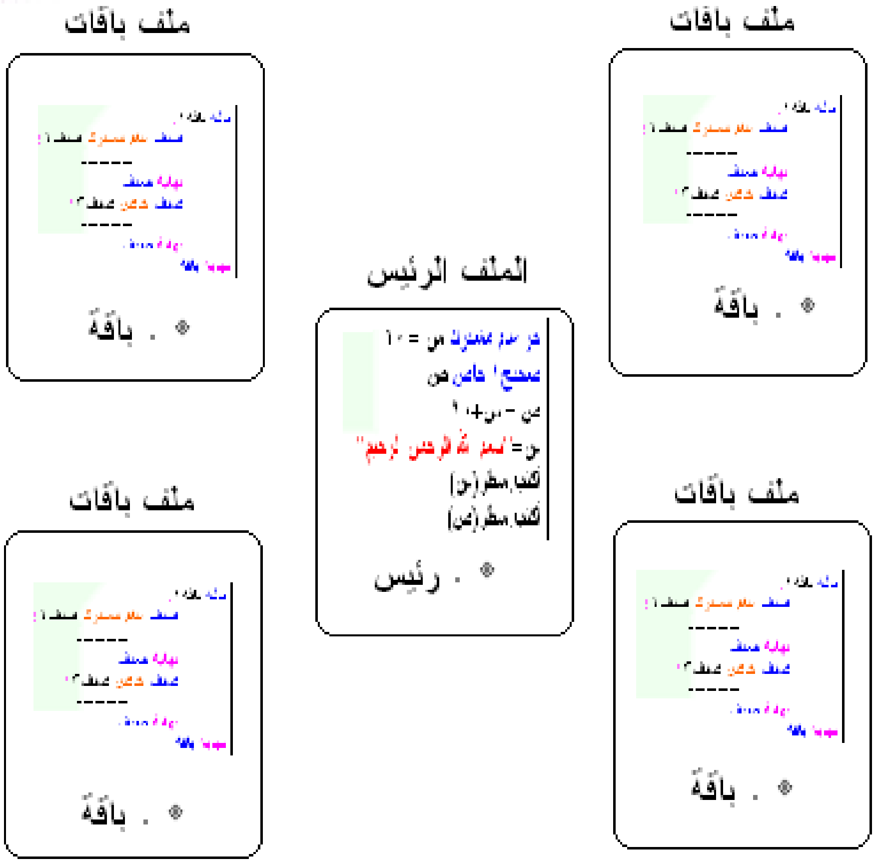
تكتب به أوامر البرنامج.

امتداده .رئيس

2- ملف باقة:

الأصناف الخارجية.

امتداده .باقة



تكوين الملف الرئيس باختصار:

- أوامر ضم المكتبات(إن وُجِدَت)
- أوامر البرنامج
- الإجراءات المنفردة (إن وُجِد أيٌ منها)

و/أو تعريفات الألقاب المنفردة (إن وُجِد أيٌ منها)

و/أو الأصناف (إن وُجِد أيٌ منها)

```
أضم الدخل.الخرج_الشاشة

رقم س= 20  ص
ص= تربيع(س)
أكتب.سطر("تربيع س = "+ص)
حساب  الحساب(ص)

إجراء (رقم التربيع) تربيع(رقم الرقم):
    التربيع = الرقم × الرقم

لقب عام نوع.الحساب:
    مفتوح  وديعة
صنف حساب:
    نص الإسم
    رقم القيمة
    \ مشيد الصنف /
    إجراء حساب(رقم قيمة):
        القيمة = قيمة
```

تكوين ملف الباقة باختصار:

- أوامر ضم المكتبات(إن وُجِدت)
- تعريفات الألقاب المنفردة (إن وُجِد أيٌ منها)
- تعريف الصِّنْف (نفس اسم الملف)

```
أضم الدخل.الخرج_الشاشة

لقب عام نوع.الحساب:
    مفتوح، وديعة

صنف حساب:
    رقم القيمة، الفائدة = 0.10
    نص الإسم
    نوع.الحساب النوع

    إجراء حساب(رقم ق):
        القيمة = ق

    صنف صنف.فرعي:
        رقم أ، ب
        إجراء إجراء.ما:
            \ أوامر الإجراء /
```

أنواع البيانات فى إبداع:

تنقسم أنواع البيانات فى إبداع إلى:

1 - أنواع بياناتٍ أصلية (المتغيرات و الثوابت).

رقم ، نص ، منطق ، حر

```
نص أ = " مرحباً "
رقم س= 20 ص
منطق ع = صحيح  ل = خطأ ك
حر ج = س+2
ج = " هذا نص ! "
```

2- أنواع بياناتٍ مركبة، تجميعةٌ من الأنواع البسيطة هي:

- الجداول: مزيجٌ من الـarrays و الـhashtables،
- الأصناف: classes،

مثال 1:

```
رقم{1} عام مشترك الأعداد{3 4}
```

مثال 2:

```
رقم{2} عام مشترك الأعداد ={ الصف.الأول={4 6 2}  الصف.الثانى={33 16 10}}
```

مثال 3:

```
رقم{2} عام مشترك الأعداد
الأعداد = {{4 6 2} {33 16 10}}
```

يمكن استخدامه كـhashtable:

مواصفات لغة إبداع:

هى لغة

بسيطة: عدد قواعدها كأقل ما يكون،

رقم س

المواصفات القياسية لها تقع في أقل من 7 ورقات قطع كبير !

سهلة و تعليمية: يمكن استخدامها كلغة تعليمية فى مجال البرمجة للأطفال الصغار بسهولة و سرعة بامتياز.

```
\ تعريف رقمين /
رقم س = 10    ص
\ الحصول علي تربيع الرقم الأول /
ص= تربيع (س)
\ طباعة الرقم س في سطر منفصل /
أكتب سطر(" س = " + س)
\ طباعة التربيع في سطر جديد /
أكتب سطر("تربيع س = " + ص)
```

هجينـة: لهـا صـفات مشتركـة تمزج بين صـفات اللغات الإجرائيـــة procedural و لغات البرمجة الكائنيـة oop، و تحاول استخلاص الأفضل من كل نوع من النوعين،

```
أضم الدخل.الخرج_الشاشة

رقم س= 20 ص
ص= تربيع(س)
أكتب.سطر("تربيع س = "+ص)
حساب  الحساب(ص)

إجراء (رقم التربيع) تربيع(رقم الرقم):
    التربيع = الرقم × الرقم

لقب عام نوع.الحساب:
    مفتوح  وديعة
صنف حساب:
    نص الإسم
    رقم القيمة
    \ مشيد الصنف /
    إجراء حساب(رقم قيمة):
        القيمة = قيمة
```

قوية: لأنها تراعى القوة فى كل مناحى تصميمها، و تعبيرٌ واحدٌ فيها يغني عن كل نظائره في اللغات الأخري !

```
بينما م في {من البداية إلي النهاية بقيمة 5}:
    أكتب.سطر("م = "+ م)

بينما م في {1 2 5 7 10}:
    أكتب.سطر("م = "+م)

بينما صحيح:
    \ نفذ أمراً ما /
    لو شرط:
        أخرج()
```

```
لو س = ص:
    أكتب.سطر("س يساوي ص")
أما لو س في {من 1 إلي 12}:
    أكتب.سطر("س في المجموعة من 1 إلي 12")
أما لو س> 12 و س<20:
    أكتب.سطر("س بين الـ 12 و الـ 20")
غيره:
    أكتب.سطر("س خارج الاحتمالات الممنوحة")
```

```
، ع  = إجراءما(5)        \.تم إهمال الخرج الأول فقط للإجراء /
،   = إجراءما(  "نص" )       \ تم إهمل جميع خروجات الإجراء /

إجراء (منطق ص  رقم ك) إجراءما(رقم   س):
    \ جسم الإجراء /
إجراء (منطق ص  رقم ك) إجراءما( نص  س):
    \ جسم الإجراء /
```

```
إجراء مخصص (رقم ص) إجراءما(رقم  س):
يخص 8086 و 80386:
    أحمل [مرجع س]، م 1
    أضع 3، م 2
    أجمع م 1، م 2
    أخزن م 2، [مرجع ص]
```

```
نداءالحالات_أضيف( ماعددالحالات('ق'))
```

آمنة و نظيفة: تختار للمبرمج أفضل الإختيارات و الأدوات و تتيحها له ليستخدمها فى برامجه.

```
صنف عام غلط1 يرث غلط:
    نص السبب
    إجراء إلى.نص:
        أكتب("هذا غلط من النوع غلط1")
    إجراء تنبيه:
        أكتب("هذا تنبيه على وقوع الغلط في البرنامج بسبب أن " + السبب)
```

```
أحاول:
    لو س > ص:
        غلط1 الغلط()
        الغلط_السبب= "المتغير س > المتغير ص"
        أعالج(الغلط)
    ع = ص - س
معالجة غلط1 غ:
    غ_تنبيه()
```

متعددة الخيوط multithreaded: أى بإمكانها إنتاج برامج ذات خيوط تنفيذ مختلفة،

تفسيرية interpreted و مترجمة compiled: أى يمكنها أن تنتج كوداً محكوماً و كوداً أصلياً باستخدام نفس النص البرمجى الإبداعى و نفس المكتبات،

مستقرة: فلا نية هناك على الإطلاق لِلإضافة إليها إلا بمعدلاتٍ بسيطةٍ جداً، و عند الحاجة الماسة لذلك، و على العموم فالتغيرات ستكون قليلة للغاية إن حدثت.

اللغات التى تم الاستفادة منها عند تصميم إبداع:



C# and Java

Matlab

C++

Visual basic

F#

ج

Fortran

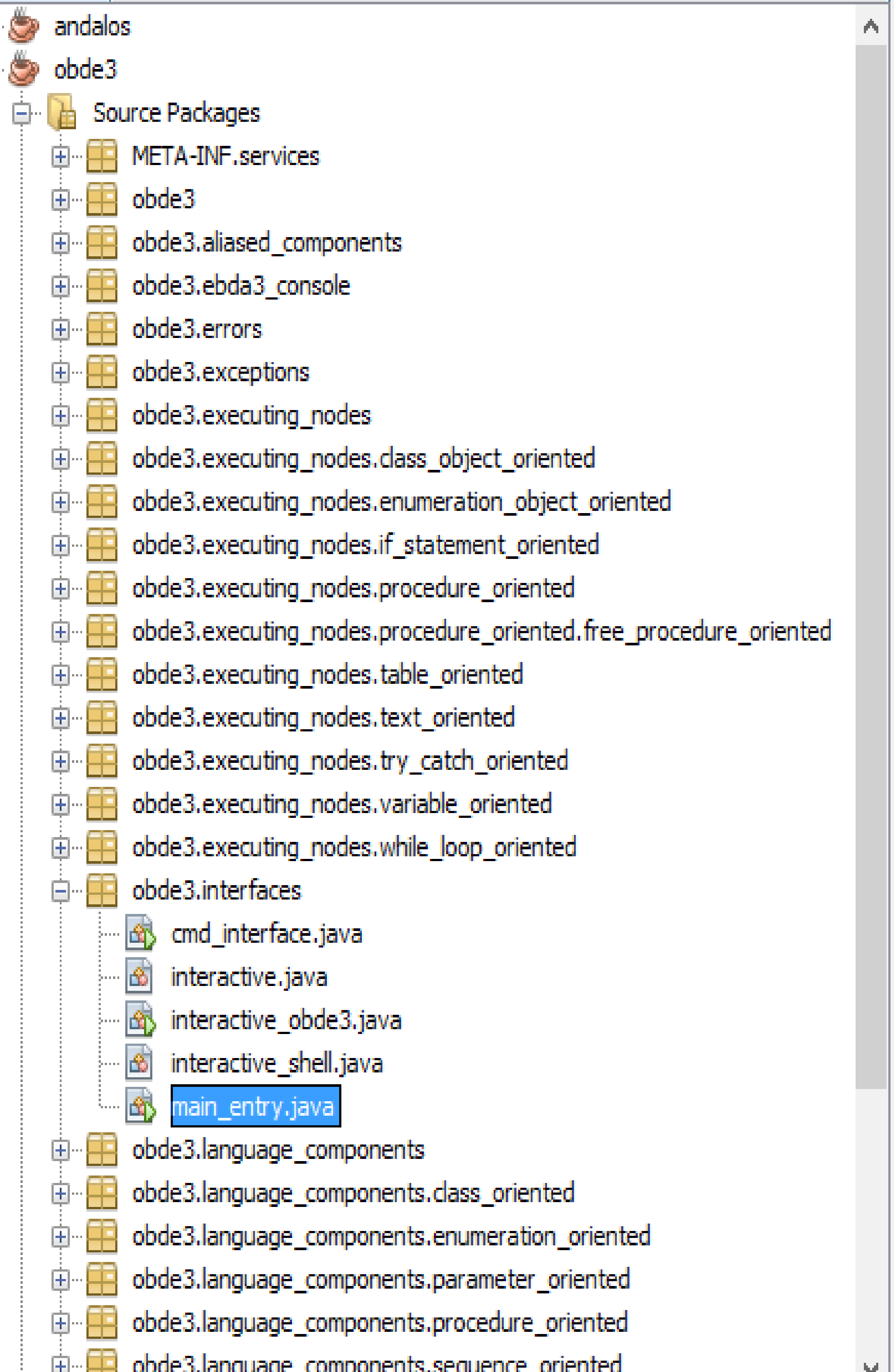
Ada

Pascal

python

Eiffel

C



أين نحن الآن؟

- مرت ستة أشهر فقط على الإطلاق الرسمي للمشروع!
- ما يقرب من 80٪ من مواصفات إبداع قد تم بناؤها.
- المُفسِّر القياسي "أُبْدِع" له واجهتان: العاديةَ و التفاعلية.
- بيئة البرمجة المتكاملة "أندلس" لها إصدارة بدائية أساسية.
- المُفسّر حوالي 35000 سطر بالjava.
- انتهى الإصدار 1.2 من كتاب (رسالة البرمجة بإبداع). و الإصدار 1.3 في الطريق بإذن الله تعالى.

رسالة البرمجة بإبداع (مانافيستو إبداع):



كتابٌ يتحدث عن:

- 1 - الواقع العلمي للامة الإسلامية.
- 2- إنتاج لغة برمجةٍ عربية.
- 3- الرد علي المعارضين.
- 4- وصف لغة (إبداع).
- 5- شرح أسس تصميم إبداع:
  - روح الإبداع.
  - مكوناتٌ رُفِض ضمها.
  - أين الإبداع فيها ؟
- 6- شرح الإصدارة رقم 1.0 من مكتبة دار الحكمة.
- النشر عند الإعلان عن مشروع البرمجة بإبداع.

أشياء لم يسعفها الوقت !

الأصناف (المجردة، العادية).

الإجراءات الحرة.

الجداول.

استخدام مكتبة الـjdk.

التحكم في خيوط التنفيذ المتعددة.

و أشياء أخري !

كيفية المساعدة

حالياً المشروع في الطّوْر الفردي، لذا فيُمكِن المساعدة من خلال الطرق التالية:

- تحميل الكتاب الرقمي (رسالة البرمجة بإبداع) الذي يُعتبَر الكتاب الرسمي للمشروع، و التحدث عنه في مدونتك و/أو موقعك و/أو حسابك الشخصي في الشبكات الاجتماعية المختلفة.

ebda3lang.blogspot.com

- تجربة اللغة بتحميل إصدار الإصدارة التجريبية الأولي من مُفسِّرها القِياسي أُبْدِع obde3 ثم إرسال التساؤلات و/أو الإقتراحات و/أو الإنتقادات و/أو المديح :) على البريد الإلكتروني لمُصمّم اللغة:

wael_hasan_87@yahoo.com

و التحدث عن التجربة مع الآخرين في المواقع الإلكترونية و غيرها، ثم نسخ المُفسِّر بشكلٍ مجانيٍ لمن يرغب في ذلك.

- الدعاية للمشروع